

U.S. Appl. No. 09/924,823  
Amendment Dated Nov. 1, 2004  
Reply to Office Action of June 30, 2004  
Docket No. BOC9-2000-0083 (218)

### REMARKS/ARGUMENTS

These remarks are submitted responsive to the office action dated June 30, 2004 (Office Action). This response is being filed with a petition for a one-month retroactive extension of time.

Before progressing to the rejections in the Office Action, a brief review of the prosecution history to this point can be beneficial. In an office action dated January 2, 2004, the Examiner indicated that claims 2 and 20 would be allowable if rewritten in independent form. To expedite prosecution, Applicants amended claims 2 and 20 in the suggested fashion and canceled claims 8-18 and 26 without prejudice or disclaimer and without discussing the merits of the rejections as noted in Applicants response dated April 30, 2004.

The Examiner then issued a final rejection against all claims in an office action dated May 18, 2004 based upon new arguments. In response, Applicants conducted a telephone interview with the Examiner on June 8, 2004, at which point, the Examiner graciously agreed to withdraw the finality of the office action dated May 18, 2004.

In response to the Office Action, and consistent with the telephone interview, Applicants respectfully reassert the claims as existing in their original state before the amendment of April 30, 2004, which was performed to expedite prosecution based upon subject matter deemed allowable, and which is now moot in light of the May 18, 2004 office action. Specifically, the previously canceled claims 2, 20, 8-18, and 26 are

U.S. Appl. No. 09/924,823  
Amendment Dated Nov. 1, 2004  
Reply to Office Action of June 30, 2004  
Docket No. BOC9-2000-0083 (218)

reasserted in their original form and independent claims 1 and 19 are reasserted in their original form.

New claims 27-30 that are dependent upon claims 1, 8, 9, and 14, respectively, have been added to clarify that the step of associating related records establishes intratable referential integrity (the term used for the disclosed concept within the title of the application) that causes the purging step to occur, as supported by page 10, line 7 to page 12, line 27. No new matter has been added as a result of these amendments.

In paragraphs 3-4, the Examiner has rejected claims 1, 3-4, 7, 19, 21-22, and 25 under 35 U.S.C. § 103(a) as being unpatentable over U.S. Patent No. 5,978,770 to Waytena, *et al.* (Waytena) in view of U.S. Patent No. 5,832,451 to Flake, *et al.* (Flake). In paragraph 5 of the Office Action, the Examiner has rejected claims 5-6, and 23-24 under 35 U.S.C. § 103(a) as being unpatentable over Waytena in view of Flake and in further view of U.S. Patent No. 5,499,359 to Vijaykumar (Vijaykumar). As per the original office action (January 2, 2004), claims 8-11, 13-16, 18, and 26 stand rejected under U.S.C. § 103(a) as being unpatentable over Waytena in view of Flake. Additionally, claims 12 and 17 stand rejected under U.S.C. § 103(a) as being unpatentable over Waytena in view of Flake, in further view of Vijaykumar.

Prior to addressing the rejections on the art, a brief review of the Applicants claimed invention is beneficial. Applicants claim a system, method, and apparatus for adding functionality to a relational database management system (RDBMS). More

U.S. Appln. No. 09/924,823  
Amendment Dated Nov. 1, 2004  
Reply to Office Action of June 30, 2004  
Docket No. BOC9-2000-0083 (218)

specifically, the present invention teaches the concept of intra-table referential integrity or referential integrity between rows of a single RDBMS table (rather than between tables which is a common functionality of a RDBMS). One result of intra-table referential integrity is that a plurality of records in a table can be grouped as a set of records, such that when one record in the set is purged from the table the grouped records within the table can be automatically purged.

Before turning to the rejections of the art, it may be helpful to elaborate upon a few RDBMS specific terms claimed by the Applicants. First, a RDBMS is a database management systems that maintains data records and indices in tables. Relationships may be created and maintained across and among the data and tables. An RDBMS publishes a plurality of functions that can be used to access and modify the data, tables, and relationships of the RDBMS.

Common published functions include commands in accordance with the Structured Query Language (SQL) that is a standard interactive and programming language for getting information from and updating information within a database. Although SQL is both an ANSI and an ISO standard, many database products support SQL with proprietary extensions to the standard language.

Referential integrity refers to a feature provided by an RDBMS that prevents users or applications from entering inconsistent data. Most RDBMS's have various referential integrity rules that can be applied to a relationship between two tables. No conventional

U.S. Appl. No. 09/924,823  
Amendment Dated Nov. 1, 2004  
Reply to Office Action of June 30, 2004  
Docket No. BOC9-2000-0083 (218)

RDBMS permits a referential integrity rule or conditions to be established between entries within a single table, which is taught by the present invention.

As an example of referential integrity between tables, suppose Table B has a foreign key (refers to a unique identifier or primary key of a record stored in a different table) that points to a field (primary key) in Table A. Referential integrity would prevent you from adding a record to Table B that cannot be linked to Table A. In addition, the referential integrity rules might also specify that whenever you delete a record from Table A, any records in Table B that are linked to the deleted record will also be deleted. This is called cascading delete, where a deletion in one table can result in N number of records in M number of tables (linked through referential integrity rules) being ultimately deleted. Finally, the referential integrity rules could specify that whenever you modify the value of a linked field in Table A, all records in Table B that are linked to it will also be modified accordingly. This is called cascading update.

As an example of intra-table referential integrity (often called intra table record associations in the specification) as noted at page 8, lines 14-28, a RDBMS can be the back-end of an airline reservation system having a front-end interface that permits airline passengers to make multiple alternative airline reservations for a single trip to accommodate for different scheduling contingencies. Intra-table referential integrity can group each of the multiple alternative airline reservations (records within a single RDBMS table) together. When one record in the set is confirmed (a triggering condition)

U.S. Appl. No. 09/924,823  
Amendment Dated Nov. 1, 2004  
Reply to Office Action of June 30, 2004  
Docket No. BOC9-2000-0083 (218)

the alternative airline reservations (related records in the table) can be purged from the RDBMS. Similar to inter-table referential integrity, intra-table referential integrity can have a cascading affect. Intra-table referential integrity in a RDBMS can provide a published functionality that can save significant development time that would have to be spent when using conventional RDBMS, where functionality to delete related records in a table would require code modifications within front-end applications and possibly in a back-end table structure and code as well.

Having described the present invention and detailed RDBMS terminology, it becomes apparent that the Applicants' invention is unrelated to Waytena, Flake, or Vijaykumar. The Applicants' invention, as noted in each independent claim and at page 8, lines 2-4, provides a method, system, and apparatus for adding functionality to a RDBMS. In contrast, Waytena provides no teachings pertaining to maintaining relationships between records and provides no teachings directed towards a RDBMS. Instead, Waytena teaches that multiple patrons having multiple personal communication devices can make reservations using a series of distributed computers (attraction computers) that are linked. Waytena's teachings are directed towards networking and distributed computing. Waytena directs no teachings towards databases in general or towards RDBMS's in particular and does not contemplate maintaining referential integrity in any fashion. As known by one of ordinary skill in the art, file management

U.S. Appl. No. 09/924,823  
Amendment Dated Nov. 1, 2004  
Reply to Office Action of June 30, 2004  
Docket No. BOC9-2000-0083 (218)

and message conveyance from a networking perspective is fundamentally different from record interdependencies within a RDBMS, the two being in non-analogous fields.

Referring to Flake, Flake teaches a travel management information system(MIS) that can utilize a database. Flake's implementation utilizes conventional database tools in implementing the disclosed MIS and are directed towards a particular implementation that relies upon a database. Flake provides no teachings or suggestions concerning expanding a RDBMS system. Flake does not contemplate intra-table referential integrity. Accordingly, Flake is an application (front end) that utilizes a conventional database (back end).

Referring to Vijaykumar, Vijaykumar describes a method for improving inter-table referential integrity, which is the conventional referential integrity that exists for a RDBMS. Vijaykumar fails to contemplate intra-table referential integrity in any fashion.

Turning to the rejections on the art, claims 1, 3-4, 7, 19, 21-22, and 25 have been rejected under 35 U.S.C. § 103(a) as being unpatentable over Waytena, *et al.* (Waytena) in view of Flake. Further, claims 8-11, 13-16, 18, and 26 have been rejected under 35 U.S.C. § 103(a) as being unpatentable over Waytena, *et al.* (Waytena) in view of Flake.

Referring to claims 1 and 19, Applicants claim:

receiving a plurality of related records;

inserting said plurality of related records into a single table of an

RDBMS;

U.S. Appl. No. 09/924,823  
Amendment Dated Nov. 1, 2004  
Reply to Office Action of June 30, 2004  
Docket No. BOC9-2000-0083 (218)

associating said plurality of related records as a set within said single table using a published function of said RDBMS; and  
responsive to a triggering condition, selectively purging particular related records of said set from said single table.

Column 10, lines 63-67 and column 23, lines 40-67 of Waytena is cited for teaching inserting related records into a single table of an RDBMS. The cited portion, however, only teaches that a virtual queue 210 can receive reservations from remotely located devices over a network. The virtual queue 210 represents a generic memory queue and is not equivalent to a single table of a RDBMS. Waytena defines the visual queue 210 as a linked list that is implementable in any fashion that queues items according to arrival time. No relationships are maintained among records in the virtual queue 210.

The Examiner admits that Waytena fails to teach associating a plurality of related records in a set within a single RDBMS table. Nevertheless, Waytena is cited for teaching the purging step. The purging step, however, has no meaning in absence to the associating step and is explicitly limited to selectively purging records based on associations within a single RDBMS table (i.e. an intra-table referential integrity delete function). That is, it is known that records can be generally removed from a queue or any other data store as taught by Waytena. It is the technique of purging records in a previously defined set from the table (intra-table referential integrity taught by the Applicants) that is novel and is not contemplated by Waytena.(or Flake).

U.S. Apph. No. 09/924,823  
Amendment Dated Nov. 1, 2004  
Reply to Office Action of June 30, 2004  
Docket No. BOC9-2000-0083 (218)

The Examiner cites column 8, lines 33 to 67 of Flake as teaching associating related records as a set of a single table. Flake provides no such teaching. Instead, Flake teaches that travel request information can be placed in a queue (again the Examiner is mistaking a queue for a RDBMS table). Flake teaches that an agent can be associated with a set of queues that designate types of tasks. Flake also teaches that current requests can be associated with passenger name records (PNR). Flake, however, makes no reference that associations are to be made within a SINGLE RDBMS table.

One of ordinary skill in the art attempting to implement the teachings of Flake would not use a single RDBMS table to store PNRs and current requests. That is, multiple tables would have to be utilized in order to assure that no functional or transitive dependencies exist between two or more nonprimary key attributes (meaning the database would be placed in third normal form or 3NF), which is common practice for those of ordinary skill with databases. In other words general principles of database construction, case tasks and passenger name records to be stored in different tables, otherwise the information would not be searchable using SQL statements and RDBMS techniques (again in accordance with 3NF). The referenced section generically discusses associating records, which would be presumed to be stored in a conventional fashion. Typically referential integrity would be maintained between linked tables in accordance with inter-table (conventional) RDBMS rules.



U.S. Appl. No. 09/924,823  
Amendment Dated Nov. 1, 2004  
Reply to Office Action of June 30, 2004  
Docket No. BOC9-2000-0083 (218)

Simply speaking, Flake does NOT teach or suggest from lines 33-67 of column 8 that related records inserted into a SINGLE RDBMS table are to be associated as a set of records in the SINGLE RDBMS table. Similarly, Flake (column 13, line 60 through column 14, line 49) fails to teach that associated records within a single RDBMS table are selectively purged. Flake provides no teachings relating to intra table record associations.

Referring to claims 2 and 20, Flake fails to contemplate that non-identified records of a set (in a SINGLE RDBMS table) can be purged. Instead, Flake uses inter-table referential integrity that is known in the art to perform a cascading delete operation.

Referring to claims 4 and 22, Waytena fails to provide any teachings about a single RDBMS table. Waytena teaches that queued items in a virtual queue 210 can be assigned a record number showing a sequence in the queue for the record. This record number or array index is unrelated to the data type for specifying related records within a RDBMS table, which is an entirely different concept. A data type is a type of record (like integer, float, string, Boolean) upon which operations (mathematical and/or comparison) can be performed. The record number of Waytena is an instance of an integer (or natural number) data type, not a new type of datatype as claimed by Applicants.

In paragraph 5 of the Office Action, the Examiner has rejected claims 5-6, and 23-24 under 35 U.S.C. § 103(a) as being unpatentable over Waytena in view of Flake and in

U.S. Appl. No. 09/924,823  
Amendment Dated Nov. 1, 2004  
Reply to Office Action of June 30, 2004  
Docket No. BOC9-2000-0083 (218)

further view of Vijaykumar. Vijaykumar, however, fails to cure the deficiencies of Waytena and Flake.

Referring to claims 5 and 23, Vijaykumar teaches that inter table referential integrity (conventional) can be disabled, thereby disassociating selected records residing in multiple RDBMS tables. Vijaykumar provides no teachings regarding disassociating selected records from a set or association defined for a single table (intra-table referential integrity - novel).

Referring to claim 7 and 24, Vijaykumar teaches a cascading delete operation among records of different tables in accordance to conventional referential integrity rules. Vijaykumar provides no teachings regarding disassociating selected records from a set or association defined for a single table (intra-table referential integrity).

Referring to claims 8, 9, 14, and 26, the Applicants claim a method and system where records are associated within a single table of an RDBMS (to establish intra-table referential integrity among related records within a single RDBMS table). For reasons stated above, neither Waytena, Flake, or combinations thereof teach or suggest (1) establishing an association among records in a single RDBMS table, or (2) selectively purging records within that single RDBMS table based upon an intra table association.

Respectfully, the Examiner is either misreading Waytena, Flake, and/or Vijaykumar, misinterpreting terms used in the claims and supported by the specification, or failing to understand general principles pertaining to a RDBMS. Specifically, neither

U.S. Appln. No. 09/924,823  
Amendment Dated Nov. 1, 2004  
Reply to Office Action of June 30, 2004  
Docket No. BOC9-2000-0083 (218)

Waytena, Flake, Vijaykumar or combinations thereof contemplate intra-table referential integrity as claimed by the Applicants (i.e. inserting related records in a single table, associating the related records as a set, selectively purging records in the set based upon a trigger – where the purged records were not specifically identified and/or where the association occurs via a published RDBMS function).

In light of the above, the 35 U.S.C. § 103(a) rejections as to claims 1-26 should be withdrawn, which action is respectfully requested. Consequently, Applicants believe that this application is now in full condition for allowance, which action is respectfully requested. The Applicants request that the Examiner call the undersigned if clarification is needed on any matter within this Amendment, or if the Examiner believes a telephone interview would expedite the prosecution of the subject application to completion.

Respectfully submitted,

Date: 1 Nov 2004



Gregory A. Nelson, Registration No. 30,577  
Brian K. Buchheit, Registration No. 52,667  
AKERMAN SENTERFITT  
Customer No. 40987  
Post Office Box 3188  
West Palm Beach, FL 33402-3188  
Telephone: (561) 653-3000